**Europäisches Patentamt**

**European Patent Office**

**Office européen des brevets**

# Bescheinigung    Certificate    Attestation

| | | |
|---|---|---|
| Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein. | The attached documents are exact copies of the European patent application described on the following page, as originally filed. | Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante. |

**Patentanmeldung Nr.    Patent application No.    Demande de brevet n°**

00200103.0

Der Präsident des Europäischen Patentamts:
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

**I.L.C. HATTEN-HECKMAN**

DEN HAAG, DEN
THE HAGUE,    26/07/00
LA HAYE, LE

EPA/EPO/OEB Form    1014    - 02.91

This Page Blank (uspto)

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

# Blatt 2 der Bescheinigung
# Sheet 2 of the certificate
# Page 2 de l'attestation

Anmeldung Nr.:
Application no.:  00200103.0
Demande n°:

Anmeldetag:
Date of filing:  13/01/00
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
Koninklijke Philips Electronics N.V.

5621 BA  Eindhoven

NETHERLANDS

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Design and real-time implementation of a low-cost noise reduction video system

In Anspruch genommene Prioriät(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat:                    Tag:                    Aktenzeichen:
State:                    Date:                    File no.
Pays:                    Date:                    Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:

/

Am Anmeldetag benannte Vertragsstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

# DESIGN AND REAL–TIME IMPLEMENTATION OF A

# LOW–COST NOISE REDUCTION VIDEO SYSTEM

*Livio Tenze, Sergio Carrato, Stefano Olivieri**

D.E.E.I., University of Trieste, via A. Valerio, 10, 34100 Trieste, Italy

email: tenze@ipl.univ.trieste.it, phone: +39 040 6767140, fax: +39 040 6763460

* Philips Research Monza, via G. & A. Philips 12, 20052 Monza (MI), Italy

A novel noise reduction system for video sequences is presented. It is based on a simple and accurate estimation algorithm, which is used to enable a filter in a suitable set of rational and median operators.

The system, which has been implemented in real time for CIF images on a commercial multimedia DSP, has been tested on both synthetic and real-world noise, and is shown to be able to effectively reduce both short- and long-tailed noise.

## Introduction

There is presently an increasing interest in digital transmission of image sequences, e.g. through the Internet. Especially in the consumer electronics area, the sources of these images, such as video-cameras, video-recorders, satellite receivers and others are affected by various types of noise. In particular, in the case of CCD and CMOS cameras, the sensor noise is usually modelled as white Gaussian [1], whereas vertical or horizontal streaks may be found in video scanned from motion picture films or played by a video cassette recorder [2], respectively.

Before storage and/or transmission, it is obviously advisable to reduce the noise level in the images, both to improve the visual appearance and to reduce the bit rate. Various algorithms have been presented in the literature for the attenuation of noises having different distributions, which are generally very complex and consequently not amenable to real time implementation in consumer equipment, or provide poor performance, typically introducing artifacts and smoothing edges.

In this paper, we propose a simple but effective noise reduction system composed by a noise estimation block and a set of nonlinear edge-preserving smoothing filters based on the rational [3] and median [4] operators, one for each type of noise; the estimation block identifies in real time the type of noise and enables the corresponding filter. The idea is that a set of simple filters, each optimised for a specific noise, can be more effective that a complex filter which has to cope with different noise statistics. Both the estimation block and the filters have a low computational

3                                                      13.01.2000

complexity and are amenable to low cost implementations.

In particular, we consider three different noise distributions: Gaussian, contaminated Gaussian [5], and long-tailed. For the latter, moreover, we also consider real world satellite receiver impulsive-like noise, characterised by short horizontal one pixel wide strips rather than by single noisy pixels [6].

The system has been implemented in real time using the Philips Trimedia DSP [7], which is a Very Long Instruction Word multimedia processor targeted to high volume consumer electronics appliances, such as set-top boxes. The current algorithm implementation allows an operating speed of 12.5 fps on CIF (352 × 288 pixel) images.

The paper is organized as follows. In the first section we briefly review the state of the art on video filtering algorithms for low cost applications, consequently restricting our attention to non-motion compensated systems. Motion-compensation based algorithms (e.g., [8, 9]), in turn, generally provide better performances at the cost of a much more complex structure, and are of interest mainly for professional applications. In Sec. 2, we describe the noise discrimination algorithm which controls the filter bank. In the third section, we present in detail the proposed set of rational and median-based filters and provide some experimental results. In Sec. 4, we describe the DSP implementation and give some detail on the solutions adopted to obtain a good processing speed. Finally, we draw some conclusions and discuss possible future work in this area.

# 1    Overview on the state of the art of non-motion compensated video filtering

Various video filtering algorithms have been presented in the literature in the recent years. Two main categories can be considered, according to the presence or the absence of motion compensation in the filter structure. In this section, we will briefly review those algorithms which do not rely on motion compensation; these systems are generally rather simple and are amenable to a hardware implementation with limited cost, and are typically targeted to the consumer application area.

4 13.01.2000

The simplest spatio-temporal filter consists in a 3-D linear FIR filter:

$$\hat{y}(i,j,k) = \sum_{p,q,r \in S} w(p,q,r)x(i-p,j-q,k-r) \qquad (1)$$

where $x(i,j,k)$ and $\hat{y}(i,j,k)$ represent the observed image and the filter output at the pixel $(i,j)$ in frame $k$, respectively, $S$ is the filter support and $w(p,q,r)$ are the filter weights. Different behaviours can of course be obtained by suitable choices of the weights; the averaging filter is the simplest form of (1). Within the area of fixed coefficient filters, an optimal solution is given by the 3-D Wiener filter [10] which however needs the *a priori* knowledge of the 3-D autocorrelation function for the original sequence and the assumption of 3-D stationarity.

In fixed linear filters, the low-pass action, which is generally required in order to reduce the noise, tends to blur edges; moreover, artifacts can be generated in case of moving objects or background. Better performances are provided by adaptive filters. An adaptive choice for the filter coefficients is made in [11]: according to the information provided by an edge detector, large weights are assigned to pixels belonging to the same object as that of the pixel to be filtered, low weights otherwise. Due to the noise sensitivity of edge detector, however, the more noise is present, the worse become the performances.

Linear IIR spatio-temporal filters have also been considered [12]:

$$\hat{y}(i,j,k) = [1 - \alpha(i,j,k)]\hat{y}_b(i,j,k) + \alpha(i,j,k)x(i,j,k) \qquad (2)$$

where $\hat{y}_b(i,j,k)$ is a prediction of the original sequence before the updating and $\alpha(i,j,k)$ controls the filter behaviour with respect to the original image and the predicted one. In [13] a 3-D Markovian sequence model is used; it turns out however that the resulting motion artifacts are very large and that consequently motion compensation has to be added. In [14] an autoregressive model is proposed to obtain $\hat{y}_b(i,j,k)$ as $\sum_{p,q,r \in S} \gamma(p,q,r)\hat{y}(i-p,j-q,k-r)$, where $\gamma(p,q,r)$ are the prediction coefficients. In [15], the filtering process consists of two steps, signal decomposition, where the stationary part of the temporal signal is extracted using order statistics, and noise reduction, using a recursive least square filter.

The adaptivity of the spatio-temporal filters of course implies a considerable computational complexity. In order to simplify the implementation step and to improve the real time perfor-

mances, a simpler choice can be to use a pure temporal filter, i.e. to restrict the region of support $S$ (see Eqn. 1) to the temporal direction only [13]. However, the noise suppression capabilities of the operator are poor due to the lack of spatial information. The filter performance can be somehow improved by increasing the window size in the temporal direction; however, in this case the sensitivity to artifacts resulting from motion within the sequence is further increased.

To overcome these limitations, an idea is to use a 1-dimensional spatio-temporal filter, i.e. to consider only one direction in the space of 3-D data. In [13], Huang and Hsu propose a filter which operates only in the direction of highest correlation. It may be noted that this approach somehow resembles those based on motion compensation. One of the major disadvantages of this method is that it is necessary to buffer several frames. Alternatively, a recursive 1-D approach can be followed: the general form is the same as (2), where however the prediction $\hat{y}_b(i,j,k)$ is obtained using only the past estimates in the temporal direction. Recursive adaptive filters have also been presented in [16, 17], in particular for camera noise reduction, which however do not reduce noise in moving areas.

Nonlinear filters, often based on order statistics, have also been used as in Arce [18] and Alp *et al.* [19]: the multistage and multilevel median filters, using both spatial and temporal information, allow to preserve the small details that a simple median operator would remove. In [13], Huang and Hsu propose a simple temporal median filter; Naqvi *et al.* in [6] propose a temporal median filter for digital television applications. In particular, in [6], it is shown that most of the temporal signal not containing noise is a root signal of the median filter: so for impulsive noise the noisy pixels are filtered whereas the noise free signal is not modified. Some problems occur when a small object travels rapidly across the screen or a scene change occurs: in this case the pixel values are not monotone in time and the performances of the median get worse. Indeed, according to Brailean *et al.* [12] the temporal median filter is not the best choice for sequences with a large amount of noise, because the temporal signals show a Gaussian behaviour for high noise levels, so that the average filter is more suitable, while the temporal median filter introduces artifacts at noisy spatio-temporal edges.

In [20], a set of nonlinear filters based on a class of Volterra expansion is presented. In [2],

6                                                    13.01.2000

morphological operators are used to treat film dirts or scratches and median operators are used to remove the line scratches caused by VCR mechanical damages. Recently, Lee and Kang [21] have proposed an extension of the 2-D anisotropic diffusion equation to a 3-D support. In [22], a noise reduction scheme for interlaced video is presented, which treats differently the higher and the lower frequencies, relying on the human vision capability; it consists of a detail preserving spatial algorithm which uses a set of contour oriented lowpass filters that are controlled using a corresponding highpass mask.

## 2   Noise estimation algorithm

In general, noise reduction filters are more effective on certain types of noise than on others. It is well known, for example, that median based operators are very efficient in case of long-tail noise [4], especially impulsive noise, while their use in case of Gaussian noise is not advisable, because they tend to generate streaking and blotching artifacts.

If a suitable set of filters can be designed, each of these optimised for a given type of noise, a good idea can be to estimate the type of noise and to automatically enable the corresponding filter. This has to be done, if possible, with low computational cost, in particular if real-time implementation for consumer applications is envisioned.

Commonly used algorithms are only able to estimate some noise features as variance or mean (e.g. [23]), while operators which are able to distinguish among several types of noise are very complex. For example, in [24] a block-based, nonlinear filtering technique based on SVD that employs an efficient method for estimating the noise power from input data is presented; however, an hypothesis of additive noise is requested and only Gaussian distributions are used in the examples. In [25], in order to detect and estimate both deterministic and random Gaussian signals in nonGaussian noise, the covariance of the latter is determined using higher order cumulants. The inverse problem is treated in [26], where signal detection and classification in the presence of additive Gaussian noise is performed using higher order statistics.

In the simple approach we propose, we use the kurtosis of the noise, defined as $k = \mu_4/\sigma^4$ (where $\mu_4$ is the $4^{th}$ central moment of the data), as a parameter with which to estimate the type

of noise distribution. The kurtosis, in fact, is well known to be related to the length of the tails of a distribution, being $k = 3$ for a Gaussian distribution, $k > 3$ for contaminated Gaussian and $k >> 3$ for impulsive noise [27].

In particular, we analyse a small part of each frame, supposing that the noise is spatially uniform; within this window, we try to extract the noise by computing the difference between these data, $y = x + n$ (where $x$ is the ideal, noise free image and $n$ is noise) and the same data filtered using a median filter:

$$z = y - \text{median}(y). \tag{3}$$

Due to the well-known noise reduction and edge preserving properties of the median filter [4], the resulting signal, $z$, is composed approximately of noise only [23], i.e. $z \simeq n$; the kurtosis $k$ is then estimated on $z$ to provide an indication on the type of noise.

Of course, $z$ does not coincide with the original noise $n$; consequently, its distribution does not coincide with that of $n$. It is very difficult to analytically evaluate the kurtosis of $z$, $k_z$, with respect to that of the true noise, $k_n$, due to the lack of independence between $y$ and $\text{median}(y)$. Some experiments have been performed, considering as $y$ one frame of the sequence *basket* plus added synthetic noise, and comparing the values of $k_z$ and $k_n$. As shown in Fig. 1, it may be seen that these values are very similar. The error between $k_z$ and $k_n$, which is slightly larger in case of Gaussian noise, is of course due to the fact that the median operator is not able to completely remove noise and preserve the signal, in particular for a short tailed noise as the Gaussian one. In any case, this error does not cause particular problems for the noise estimation. Indeed, it may be seen from the figure that, for reasonable values[1] of the noise variance (in case of Gaussian or contaminated Gaussian) or of percentage of corrupted pixels (for impulsive noise), the operator allows to correctly discriminate the types of noise, using two suitable thresholds.

Experimental results have shown that real impulsive-like noise is also recognised as impulsive noise. For a real sequence (500 frames) obtained from a satellite receiver we obtained $k_z = 22.9 \pm 6.1$. As described in the next section, in case of long-tailed noise we use a simple median

---

[1]Levels of noise outside the range considered in Fig. 1 yield images which are either virtually noiseless or practically useless, respectively.

8                                          13.01.2000

filter which is effective both for single noisy pixels and for horizontal and vertical streaks, so that there is no need to distinguish between ideal and real impulsive noise.

As already mentioned, only a small part of each frame (a $3 \times 3$ pixel subimage) is considered, in order to keep the computational load per frame low. Being a stable estimate needed, we have to perform a long-term analysis by cumulating the data for many frames before actually computing $k$. In order to estimate the variance of our estimator, we performed several tests on a real-world image and added noise, using several total numbers of pixels. As can be seen in table 1, an estimate over 900 pixels (i.e. over 100 frames) has a reasonably low variance. As an example, in Fig. 2 the distributions over 500 trials of our estimate for 900 pixels are reported in case of Gaussian, contaminated Gaussian and impulsive noise; it can be seen that there is no overlapping among the queues, so that it is actually possible to correctly discriminate the various noise types using two thresholds, 6 and 15. Although with other sequences a small overlapping between the tails of two adjacent distributions may appear, it may be noted that the consequent erroneous classification is not critical, because it only implies a non optimal noise filtering action for a short duration, i.e. until the following estimation is performed (typically a few seconds, as will be shown later).

## 3   The filter bank for noise smoothing

Edge preserving noise reduction can be achieved using spatio-temporal rational and median based filters. For what concerns the former ones, in fact, it is well known that they can effectively distinguish between details and homogeneous regions by modulating their overall lowpass behaviour according to the differences of suitably chosen pixels [3], so that noise is significantly reduced while details are not blurred. Their effectiveness has been shown on various types of noise, including Gaussian [3], contaminated Gaussian [28], and speckle [3, 29]. On the other hand, the good performances of median based filters on impulsive, and in general on long-tailed, noise are well established [4].

In order to be able to treat different kinds of noise, a set of three different filters has been designed; their operation is automatically controlled by the noise discriminator described earlier.

9                                    13.01.2000

Their support has been restricted to two temporally adjacent frames only, in order to keep the computational complexity low.

The filters for Gaussian and contaminated Gaussian we propose share the same structure, constituted by the sum of a spatial and a temporal filtering part. More precisely, with reference to Fig. 3, each filter output $y_0$ is computed as

$$y_0 = x_0 - f_{spatial} - f_{temp}$$

with [3]

$$f_{spatial} = \sum_{i,j \in I} \frac{-x_i + 2x_0 - x_j}{k_s(x_i - x_j)^2 + A_s}, \qquad (4)$$

where $x_0$, $x_i$ and $x_j$ are pixel values within the mask ($x_0$ being the central one), $i, j \in I$ describe the set of spatial filtering directions shown in Fig. 4, and $k_s$ and $A_s$ are suitable filter parameters; $f_{temp}$, which refers to the temporal part of the filter, has a similar form, although operating also on pixels of the previous frame, and is described later in this section.

It may be seen that the spatial filter is able to distinguish between homogeneous and detailed regions, in order to reduce noise while maintaining the figure details. In fact, if the mask lies in a homogeneous region, the pixel differences $(x_i - x_j)^2$ which appear at the denominator are small, and the high-pass component present at the numerator, which is subtracted from $x_0$, gives an overall lowpass behaviour. In turn, if the same differences have a large value, an edge is supposed to be present, and the filter leaves the pixel unchanged in order not to blur the detail.

The temporal part exploits the same principle of detail sensitive behaviour, and for Gaussian noise the form is similar to that of the spatial part:

$$f_{temp}^{(gauss)} = \sum_{i \in J} \frac{-x_i^p + x_0}{k_{t1}(x_i^p - x_0)^2 + A_{t1}}, \qquad (5)$$

where $i \in J$ describes a set of temporal filtering directions as the one shown in Fig. 5, the superscript $p$ refers to pixels belonging to the previous frame, and $k_{t1}$ and $A_{t1}$ are suitable filter parameters. The situation is slightly more complicated for contaminated Gaussian noise. In this case, in fact, details and noise are more difficult to discriminate, because the pixel noise level can be large (due to the rather long tails of the distribution), and less information with respect to the spatial case is available; more precisely, due to the limited temporal size of the filter support (only

10                                     13.01.2000

two frames), pixels are available only at one (temporal) side of $x_0$ (viceversa, in the spatial part of the filter, pixels both at the right and at the left of $x_0$, or both on top of and below, are available) so that the simple denominator of the spatial part does not allow to distinguish between a single noisy pixel and the edge of an object. We define

$$f_{temp}^{(cont\,gauss)} = \sum_{i \in J} \frac{-x_i^p + x_0}{[k_{t2}(x_i^p - x_0)^2 + k_{t3}(x_i^p - x_i)^2]/2 + A_{t2}} \tag{6}$$

where $k_{t2}$, $k_{t3}$, and $A_{t2}$, are suitable filter parameters. In this case, the pixels at the denominator, which control the "strength" of the low-pass action, are three instead of two: $x_i$, $x_i^p$ and $x_0$. In fact, as already mentioned, it is not advisable to use the same control strategy as for Gaussian noise: the difference $(x_i^p - x_0)$ may be large due to a noise peak, instead of an edge, with consequent loss of the noise filtering action. In turn, if the same difference is "corrected" by averaging with another difference, i.e. $(x_i^p - x_i)$, the denominator remains low also in presence of isolated noisy pixels, and the desired lowpass behaviour is obtained.

In order to treat effectively the impulsive noise, we use a simple median filter. In particular, we are interested in removing both ideal impulsive noise (single noisy pixels) and real world impulsive-like noise made of horizontal one pixel wide strips. A simple mask which is appropriate in these two cases is the 5 element x-shaped one [4]. In fact, both types of noise affect only one pixel out of 5 in the mask, so that the noisy element is easily removed by the median operator. It may be noticed, incidentally, that one pixel wide vertical strips, which may be found in video obtained from motion picture films [30], can also be effectively removed by this filter; of course, a larger support is required in order to remove wider strips. A fast implementation of the median filter [31] has been used in order to keep low the computational cost, which otherwise can be rather high also in case of median operators with reasonably small masks. Finally, it may be worthwhile to note that, once the impulsive noise type has been detected, there is no reason to resort to sophisticated filtering techniques as e.g. those used in [18] or [32] (which had to cope with contaminated Gaussian and constant impulse speckle); the simple median we use is computationally cheaper and, as it will be shown in the following section, yields even better results.

## 3.1  Experimental results

Some experimental results are shown in Tab. 2 and Fig. 6, 7, 8, where the real world sequence *basket* has been considered, and synthetic noise has been added [2].

Comparisons are presented with the performances provided by the MMF [18] and the Alp's ML3D method [32]. Only these algorithms have been chosen among those cited in Sec. 1 because they are roughly comparable to our one in term of computational complexity.

In order to obtain an objective measure of the noise reduction, we use the Mean Square Error parameter, defined as

$$MSE = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} (y_{ij} - x_{ij})^2}{M \cdot N}$$

where $y_{ij}$ and $x_{ij}$ denote the pixels in the filtered and the original frame and $M$ and $N$ are the image dimensions. In order to allow a subjective evaluation, some processed images are also reported in the figures. It may be seen that the proposed operator is very effective in suppressing the various types of noise. It may be noticed that the results with the rational operators in Tab. 2 are better than those obtained with the multidimensional median operators, which are designed to remove contaminated Gaussian and speckle noise. Moreover, it has to be remembered that our operator uses only 2 frames, while both MMF and ML3D need 3 frames: even though the price of memories is decreasing, the cost of an extra frame buffer can be significant in cost sensitive areas as, for example, in consumer applications.

During the system design phase, the parameters $k$ and $A$ have been chosen by applying the filter to several video sequences with added noise and by minimising the MSE between the filter output and the original, supposed noise-free, image. We performed the minimisation process using the BFGS Quasi-Newton method with a mixed quadratic and cubic line search procedure. The dependence of the parameter values on the noise level has been investigated. In particular, in Fig. 9 the values found according to different levels of noise variance (for Gaussian and contaminated

---

[2]In particular, contaminated Gaussian noise has been generated according to [5]

$$\nu \sim (1 - \lambda)\mathcal{N}(0, \sigma_n) + \lambda \mathcal{N}(0, \frac{\sigma_n}{\lambda})$$

where $\mathcal{N}(\mu, \sigma_n)$ is a normal Gaussian distribution with mean $\mu$ and variance $\sigma_n$. The $\lambda$ parameter is used to vary the form of the distribution from pure Gaussian ($\lambda = 0$) to strongly contaminated Gaussian ($\lambda \simeq 0.1$).

Gaussian) are shown. It may be seen that in the first two cases either stable values are obtained, or a definite trend can be found, so that, once the noise variance is estimated[3], the correct value of the parameters can be easily obtained by e.g. piecewise linear interpolation from a look-up table.

We also analysed the dependency of the parameters value on the training set, by testing the algorithm on a sequence different from the one used for the optimisation. As an example, in Fig. 10 the MSE is reported for several frames of the sequence *basket*, using the parameters obtained minimising the error on a small segment of three different sequences (namely, *basket*, *flower*, and *foreman*). It may be noticed that performances are only slightly better (in particular for Gaussian noise) when training and test data match.

Finally, in Fig. 11, results with real impulsive-like noise are reported, again in comparison with the two other filters considered above. Also in this case, good performances are provided by our operator.

## 4  DSP implementation

The system has been implemented on the Philips Trimedia DSP mounted on a board hosted by a PC, and a processing speed of 12.5 fps for CIF images has been obtained, thanks to various optimisations and simplifications which have been adopted.

The Trimedia processor allows to exploit a set of *custom operations* in order to optimise the speed of data processing: these are specialised high level operations designed to significantly improve performance in typical multimedia applications. When properly included into the application source code, custom operations permit to take full advantage of the highly parallel TM1000 microprocessor architecture. *Loop optimisation* techniques, which move critical code off the control flow path to reduce inner loops to a single decision tree, restricted pointers and profiling[4] procedures [33] have also been used.

The real-time evaluation of noise, needed in order to cope with possible noise distribution changes, is obtained using, as already mentioned, a long-term analysis; in particular the compu-

---

[3]It has to be noted that the computation of the noise variance is also needed by the noise discriminator algorithm.

[4]With simulation and profiling it is possible to obtain useful information about the program code structure. Statistical evaluations can then be used in a second compilation to optimise the resulting code.

tation is performed every 100 frames, which implies that, at 12.5 fps, the $k$ value is updated every 8 seconds. Using a 900 element circular array where, for each frame, the values of the 9 pixels in the mask are written, it is possible to update $k$ more frequently, e.g. every 25 frames (i.e. every 2 seconds).

For the rational operators, in order to avoid divisions, we use look-up tables. In particular, for the spatial part of Eqn. 4, we need to evaluate the denominator and to compute the fraction $1/(k_s(x_i - x_j)^2 + A_s)$. The values $x_i$ and $x_j$ are included in the range $0 - 255$, so that the value $|x_i - x_j|$ may be used to access a 256 element look-up table, which returns the inverse of the denominator; the latter is then multiplied by the numerator. Similar considerations apply to the other rational functions used.

In order to minimise the number of memory accesses, we have also quantised the look up tables. This is important because, in modern DSPs, speed is often limited by the cache, so that if a very large look-up table occupies most of it, the speed of data and/or code loading from memory is significantly reduced. Both objective results based on the MSE (see Tab. 3) and subjective observations show that a quantisation factor equal to 4 does not significantly decrease the image quality. In turn, quantisation is crucial for the temporal part of the filter for contaminated Gaussian noise, which requires the evaluation of two differences (Eqn. 6); in this case, it should be necessary to use 2-dimensional $256 \times 256$ (instead of $64 \times 64$) look up tables. Moreover, it may be noticed that quantisation using a factor 4 is easily obtained using a simple operation of bit shift. Such an implementation allows to process one frame within 7 Mcycles.

For the x-shaped median operator, as already mentioned we have used the efficient algorithm proposed in [31], which requires less than 10 Mcycles per frame.

## 5   Conclusions

We have presented a simple but effective noise reduction algorithm for image sequences. It is based on a set of rational filters and on a simple algorithm which estimates the type of noise present in the images. Experimental results show good performances for various types of noise, while the simple structure allows a low-cost real time implementation. In particular, the algorithm has

14                                          13.01.2000

been implemented on a commercial multimedia DSP, the Philips Trimedia TM1000, where 12.5 fps have been achieved for CIF format images.

The current algorithm implementation can be seen as the starting point for the design of DSP libraries. These libraries could be statically or dynamically allocated in the code memory of multimedia equipment built around DSP processors similar to the one used, and activated in all the application whenever this method is valuable.

# 6   Acknowledgements

15                                              13.01.2000

# References

[1] Glenn E. Healey and Raghava Kondepudy, "Radiometric CCD camera calibration and noise estimation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 3, pp. 267–276, Mar. 1994.

[2] N.R. Harvey and S. Marshall, "Non-linear image processing for digital TV," Final Report on EPSRC Grant No. GR/K49867, 1999.

[3] Giovanni Ramponi, "The rational filter for image smoothing," *IEEE Signal Processing Letters*, vol. 3, no. 3, pp. 63–65, Mar. 1996.

[4] I. Pitas and A. N. Venetsanopoulos, *Nonlinear digital filters*, Kluwer Academic Publishers, Boston MA(USA), 1990.

[5] Moncef Gabbouj and Ioan Tabus, "TUT noisy image database v.1.0," Esprit NAT document, Jan. 1995.

[6] S.S.H. Naqvi, N.C. Gallagher, and E.J. Coyle, "An application of median filters to digital television," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 2451–2454, 1986.

[7] "http://www.trimedia.philips.com," .

[8] K. J. Boo and N. K. Bose, "A motion–compensated spatio–temporal filter for image sequences with signal–depend noise," *Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 3, pp. 287–298, June 1998.

[9] D. S. Kalivas and A. A. Sawchuck, "Motion compensated enhancement of noisy image sequences," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 2121–2124, Apr. 1990.

[10] M. K. Ozkan, A. T. Erdem, M. I. Sezan, and A. M. Tekalp, "Efficient multiframe Wiener restoration of blurred and noisy image sequences," *IEEE Trans. on Image Processing*, vol. 1, pp. 453–476, Oct. 1992.

[11] C. B. Dekker, A. J. E. M. Janssen, and P. J. van Otterloo, "The contour plot method for noise reduction in digital video," *Acta Electronica*, vol. 27, no. 1-2, pp. 119-131, 1985.

[12] James C. Brailean, Richard P. Kleihorst, Serafim Efstratiadis, Aggelos K. Katsaggelos, and Reginald L. Lagendijk, "Noise reduction filters for dynamic image sequences: A review," *Proceedings of the IEEE*, vol. 83, no. 9, pp. 1272-1292, Sept. 1995.

[13] T. S. Huang, *Image Sequence Analysis*, Springer-Verlag, Berlin, 1981.

[14] A. K. Katsaggelos, J. N. Driessen, S. N. Efstradiadis, and R. L. Lagendijk, "Temporal motion compensated noise filtering of image sequences," *SPIE Proc. Vis. Comm. and Image Process.*, pp. 61-70, Nov. 1989.

[15] R. P. Kleihorst, G. de Haan, R. L. Lagendijk, and J. Biemond, "Motion compensated noise filtering of image sequences," *Proc. EUSIPCO-92*, pp. 1385-1388, Aug. 1992.

[16] R. H. McMann et al., "Digital noise reducer for encoded NTSC signals," *SMPTE J.*, vol. 87, pp. 129-133, Mar. 1978.

[17] T. J. Dennis, "Nonlinear temporal filter for television picture noise reduction," *Proceedings of the IEEE*, vol. 127, no. 2, Apr. 1980.

[18] Gonzalo R. Arce, "Multistage order statistic filters for image sequence processing," *Transactions on Signal Processing*, vol. 39, no. 5, pp. 1146-1163, May 1991.

[19] M. A. Alp and Y. Nuevo, "3-dimensional median filters for image sequence processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 4, pp. 2917-2920, May 1991.

[20] C. L. Chan, A. K. Katsaggelos, and A. V. Sahakian, "Linear quadratic noise smoothing filters for quantum-limited images and image sequences," *IEEE Trans. on Image Processing*, vol. 4, Sept. 1995.

[21] Suk Ho Lee and Moon Gi Kang, "Spatio-temporal video filtering algorithm based on 3-D anisotropic diffusion equation," *Proc. 1998 International Conference on Image Processing*, vol. 2, pp. 447-450, 1998.

[22] Klaus Jostschulte and Aishy Amer, "A new cascaded spatio–temporal noise reduction scheme for interlaced video," *Proc. International Conference on Image Processing*, vol. 2, pp. 493–497, 1998.

[23] S. I. Olsen, "Estimation of noise in images: An evaluation," *CVGIP*, vol. 55, no. 4, pp. 319–323, July 1993.

[24] Konstantinos Konstantinides, Balas Natarajan, and Gregory S. Yovanof, "Noise estimation and filtering using block-based singular value decomposition," *IEEE Trans. on Image Processing*, vol. 6, no. 3, pp. 479–483, Mar. 1997.

[25] Brian M.Sandler, Georgios B. Giannakis, and Keh-Shin Lii, "Estimation and detection in nonGaussian noise using higher order statistics," *IEEE Trans. on Signal Processing*, vol. 42, no. 10, pp. 2729–2741, Oct. 1994.

[26] Georgios B. Giannakis and Michail K. Tsatsanis, "Signal detection and classification using matched filtering and higher order statistics," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 38, no. 7, pp. 1284–1296, July 1990.

[27] E.Lloyd, *Handbook of applicable mathematics*, John Wiley & Sons Ltd., New York, 1980.

[28] Francesco Cocchia, Sergio Carrato, and Giovanni Ramponi, "Design and real-time implementation of a 3-D rational filter for edge preserving smoothing," *IEEE Trans. on Consumer Electronics*, vol. 43, no. 4, pp. 1291–1300, Nov. 1997.

[29] G. Ramponi and C. Moloney, "Smoothing speckled images using an adaptative rational operator," *IEEE Signal Processing Letters*, vol. 4, no. 3, Mar. 1997.

[30] Robin D. Morris, W.J.Fitzgerald, and A.C.Kokaram, "A sampling based approach to line scratch removal for motion picture frames," *IEEE Trans. on Image Processing*, vol. 1, pp. 801–804, Sept. 1996.

[31] Nicolas Devillard, "Fast Median Search: an ANSI C implementation," http://www.eso.org/ ndevilla/median/.

[32] B. Alp, P. Haavisto, T. Jarske, K. Oistaemoe, and Y. Neuvo, "Median-based algorithms for image sequence processing," in *Proc. SPIE Visual Communications and Image Processing,* Lausanne, Switzerland, Oct. 1990, pp. 122–133.

[33] Philips Semiconductors, "Trimedia SDE cookbook, optimizing trimedia applications," 1997, Part 4.

19                                          13.01.2000

## Table and figure captions

Table 1: Mean and standard deviation of the estimator of the kurtosis for different numbers of pixels, in presence of different types of noise.

Table 2: Performance comparison in terms of MSE for three different algorithms and three types of noise, for a frame of the sequence *basket*.

Table 3: Effect of the quantisation of the look-up tables on the filters performances, on an image of the sequence *basket* with quantisation factor $Q = 4$.

Figure 1: Local kurtosis $k$ for three different types of noise, for the original noise (+) and for the noise $z$ estimated using Eq. 3 ($\cdot$). It may be seen that the kurtosis computed on $z$ accurately approximates the actual value; moreover, the ranges of $k$ for the three cases are well separated.

Figure 2: Distributions (over 500 tests on 900 pixels) of the estimator of the kurtosis for (left) Gaussian with zero mean and $\sigma^2 = 200$, (center) contaminated Gaussian with zero mean, $\lambda = 0.1$ and $\sigma_n^2 = 1000$, and (right) impulsive noise with 5% of noise density.

Figure 3: Masks for the proposed rational spatio-temporal filters. The filters use suitable combinations of pixels in the current frame (as $x_i$ and $x_j$ in the figure) and in the previous one (as $x_i^p$ in the figure).

Figure 4: Directions considered in the spatial part of the rational filters: a) horizontal; b) vertical; c) and d) diagonal.

20 13.01.2000

Figure 5: Example of direction used by the temporal part of the rational filters, for (top) Gaussian, and (bottom) contaminated Gaussian noise. There is a total of 9 possible directions, according to the possible positions of $x_i^p$ (and $x_i$) on the respective masks; only one direction has been drawn for the sake of clarity.

Figure 6: Performance of three different spatio-temporal filters in case of Gaussian noise: a) original image with noise; b) proposed filter; c) MMF; d) ML3D.

Figure 7: Performance of three different spatio-temporal filters in case of contaminated Gaussian noise: a) original image with noise; b) proposed filter; c) MMF; d) ML3D.

Figure 8: Performance of three different spatio-temporal filters in case of impulsive noise: a) original image with noise; b) cross median; c) MMF; d) ML3D.

Figure 9: Dependence of rational filters parameters values on the noise level used during optimisation. From top to bottom: Gaussian and contaminated Gaussian. The values are normalised to the typical values of $k$ and $A$.

Figure 10: Dependence of the algorithm performance on the parameters value obtained using different training sets. Results shown refer to the system tested on the sequence *basket* using values obtained optimising the parameters on three different sequences.

Figure 11: Performance of three different spatio-temporal filters in case of real-world impulsive noise: a) original image with noise; b) proposed filter; c) MMF; d) ML3D.

CLAIMS:

1.        Method of noise filtering a signal, the method comprising:

identifying a type of noise in the signal;

controlling a plurality of noise filters depending on the type of noise.

5    2.        Device for noise filtering a signal, the device comprising:

means for identifying a type of noise in the signal;

means for controlling a plurality of noise filters depending on the type of

noise.

10    3.        Video system comprising:

means for obtaining an image signal,

a device as claimed in claim 2 for noise filtering the image signal.

## Tables

| Number of pixels | Gaussian | Cont. Gaussian | Impulsive |
|---|---|---|---|
| 400 | 4.38±0.53 | 9.75±1.39 | 35.01±11.38 |
| 900 | 4.39±0.39 | 10.84±0.93 | 35.64±7.97 |
| 2500 | 4.43±0.22 | 9.97±0.53 | 33.54±3.85 |

Tab. 1

| Filter | Gaussian | Cont. Gaussian | Impulsive |
|---|---|---|---|
| Original plus noise | 198 | 2285 | 1055 |
| Proposed | 84.86 | 308 | 58 |
| MMF | 125 | 595 | 70 |
| ML3D | 123 | 401 | 77 |

Tab. 2

| Implementation | Gaussian | Contaminated Gaussian |
|---|---|---|
| without quantisation | 81.24 | 307.10 |
| with quantisation | 83.55 | 363.16 |

Tab. 3
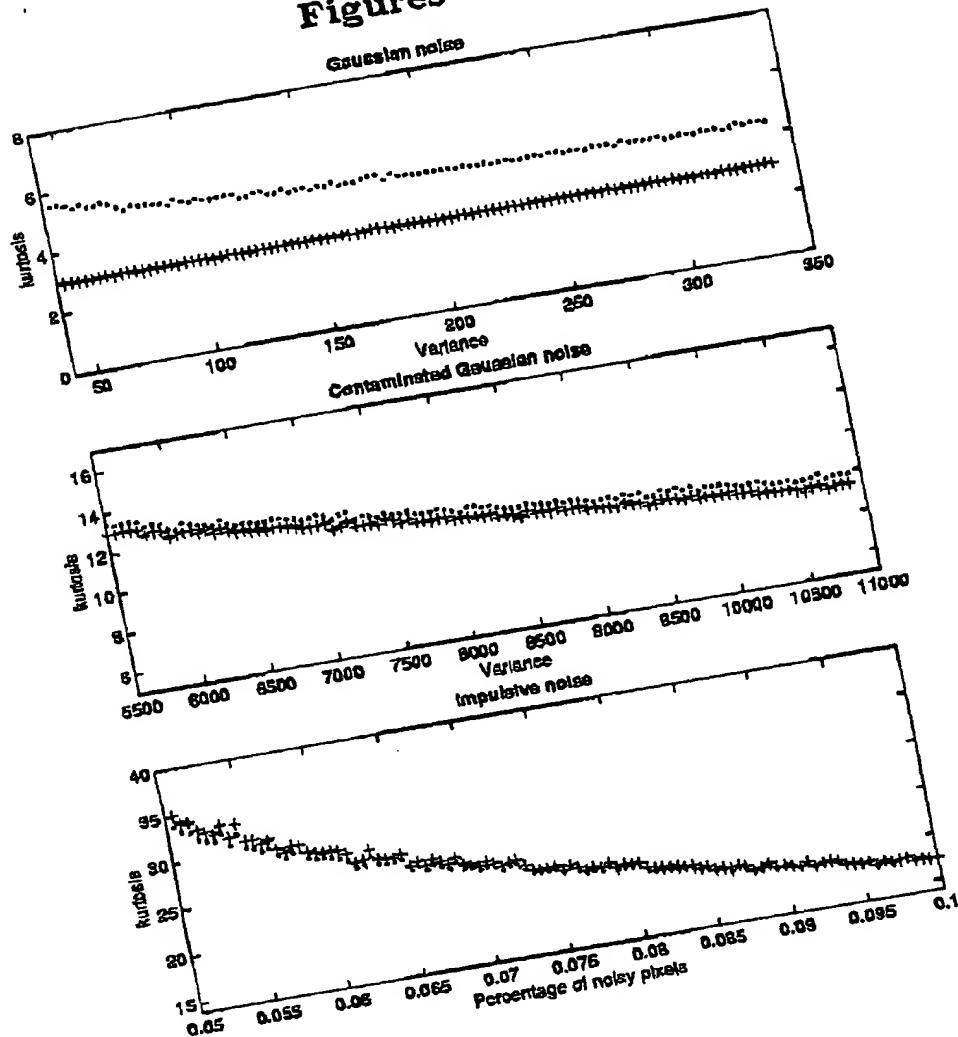
PH-IT000002

# Figures

Fig. 1
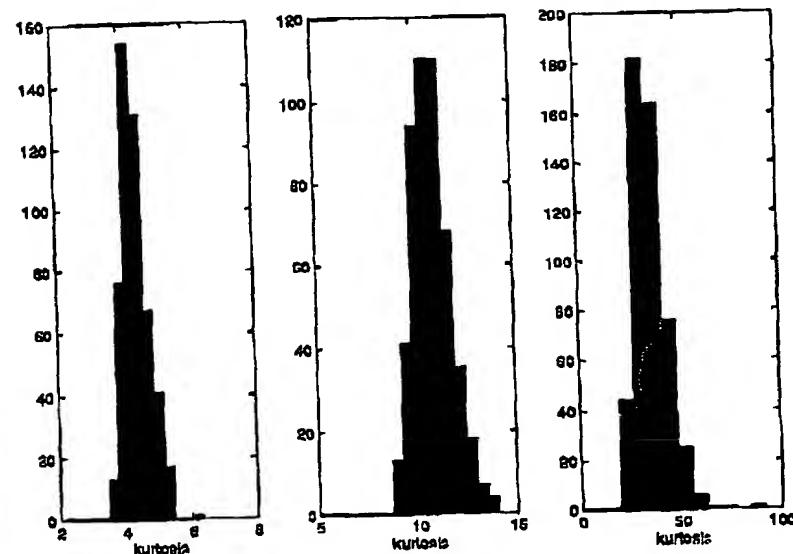
Fig. 2
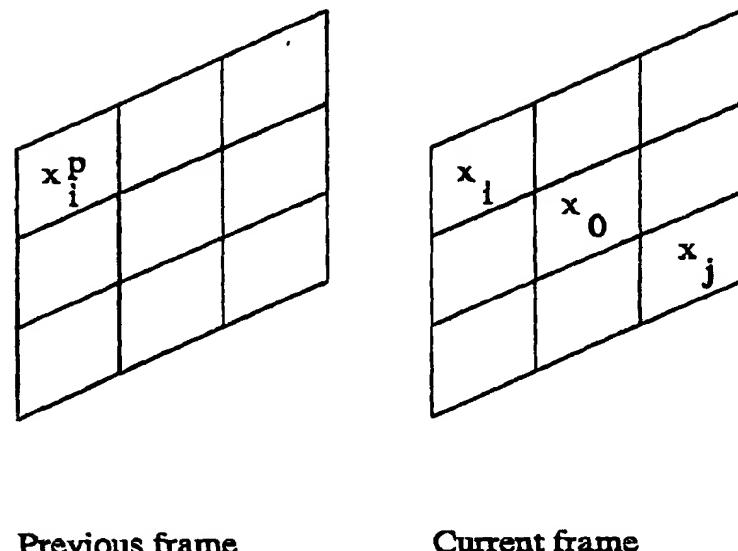
Previous frame Current frame

Fig. 3
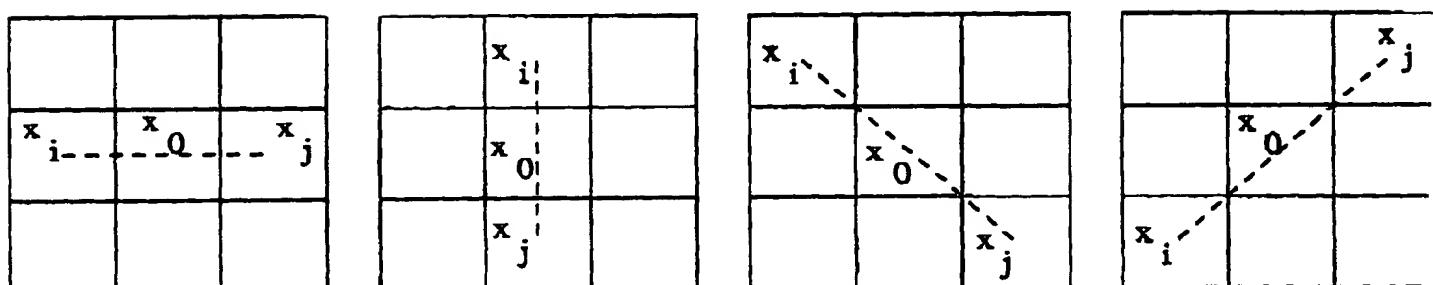


Fig. 4
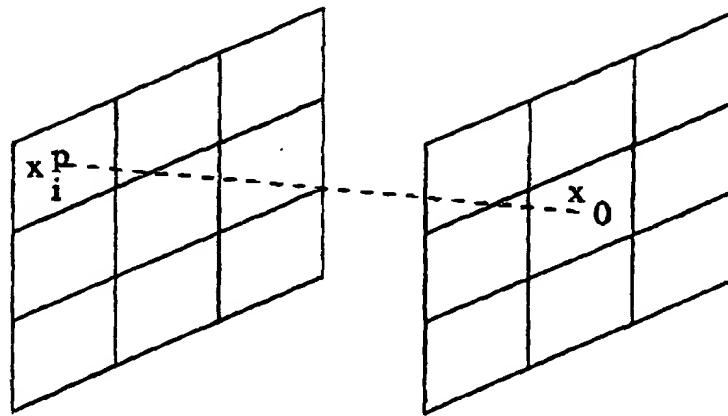
PH-IT000002

Previous frame        Current frame
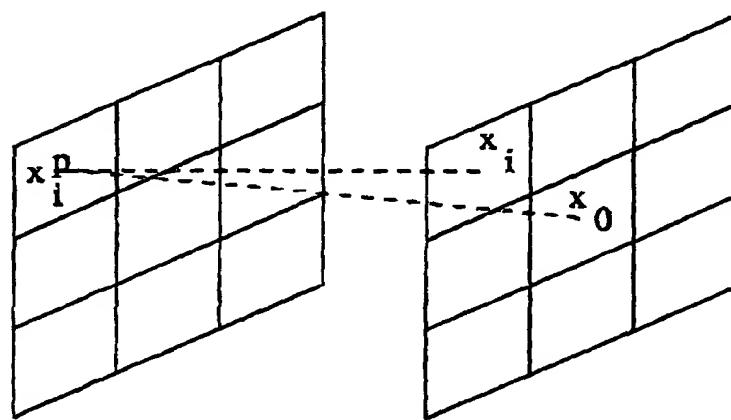
Previous frame        Current frame

Fig. 5

PH-IT000002

a)

b)

c)

d)
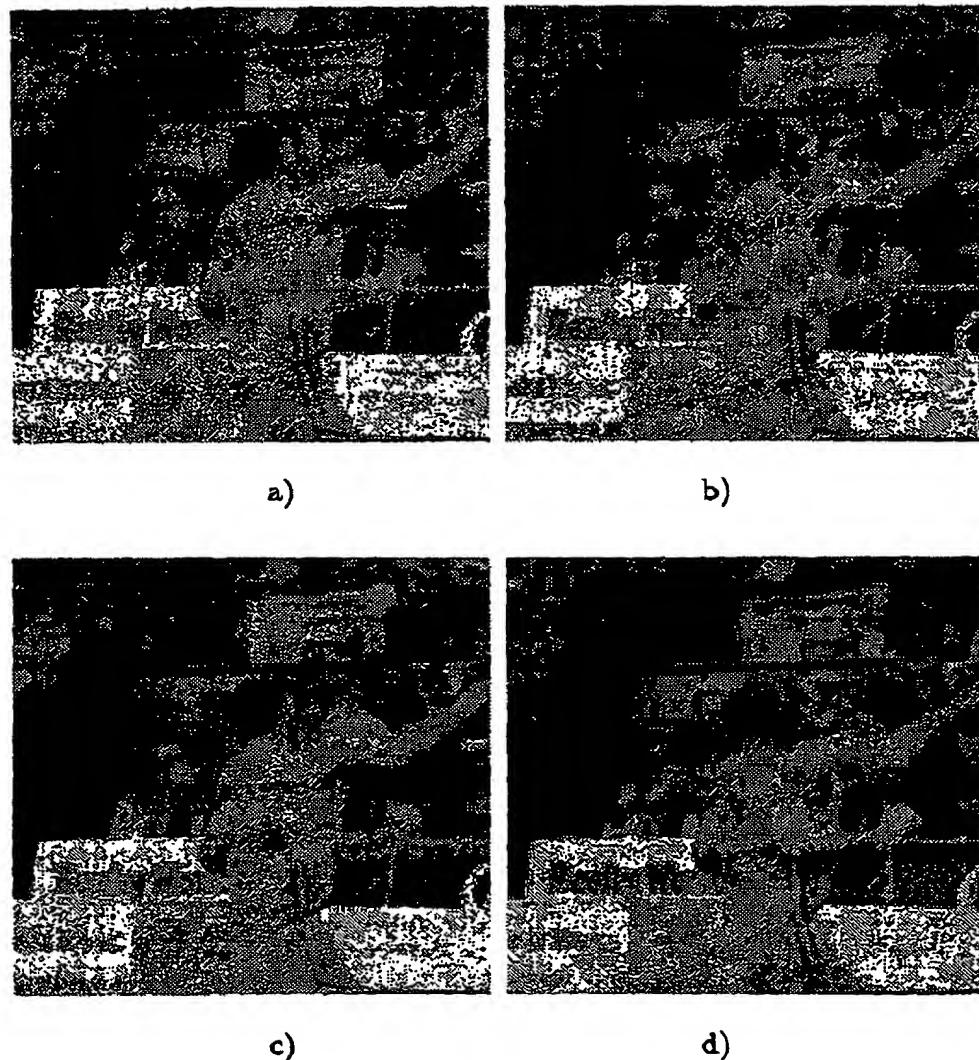
Fig. 6

BEST AVAILABLE COP

a)                                        b)

c)                                        d)

Fig. 7

BEST AVAILABLE COPY

a)　　　　　　　　b)

c)　　　　　　　　d)

Fig. 8

BEST AVAILABLE COPY

PH-IT000002
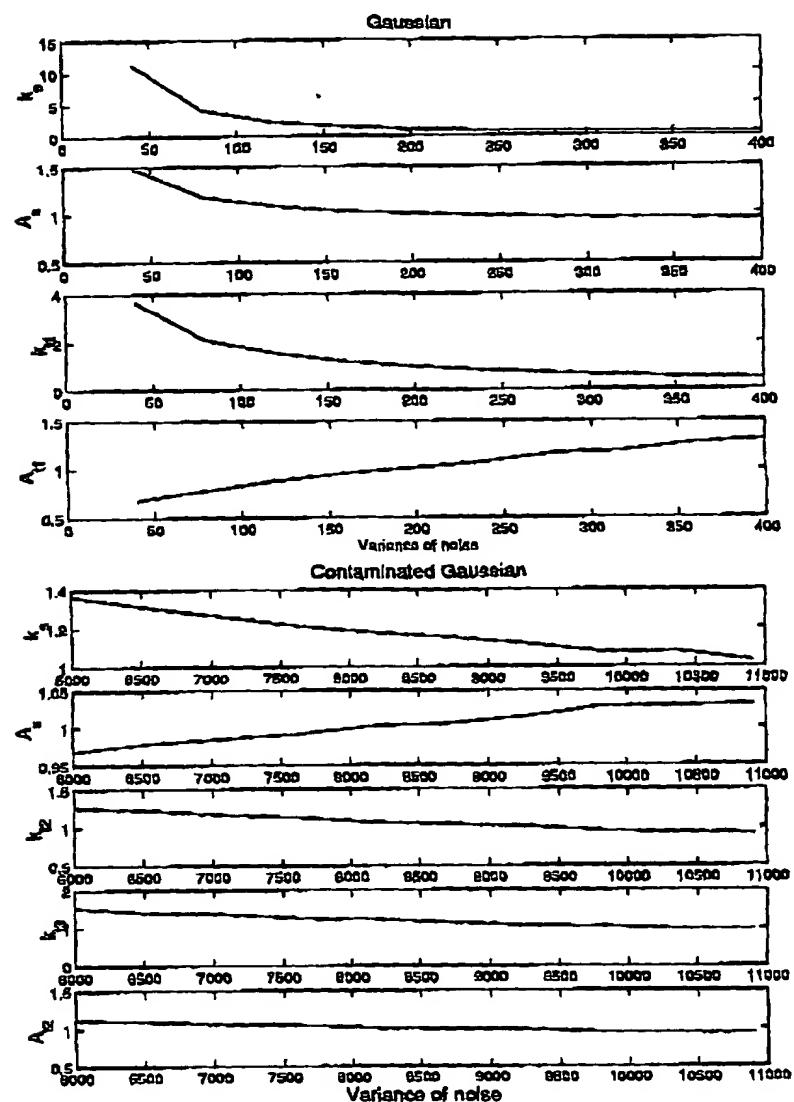
Fig. 9
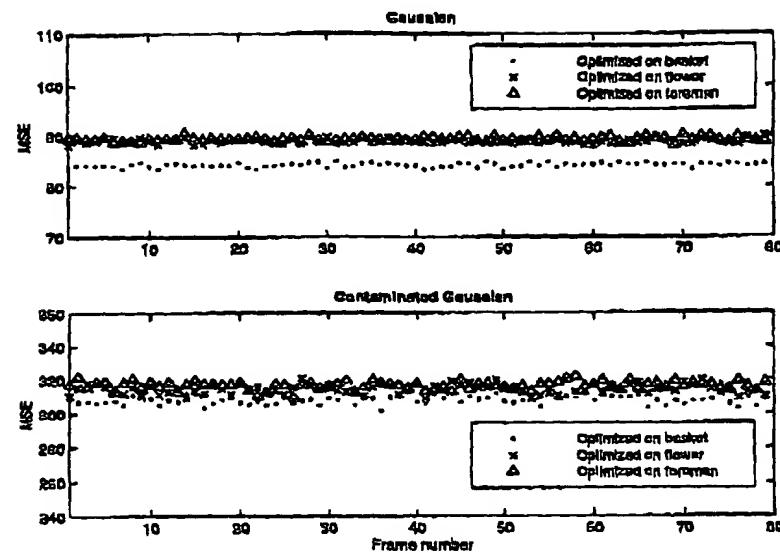
PH-IT000002

Fig. 10

PH-IT000002
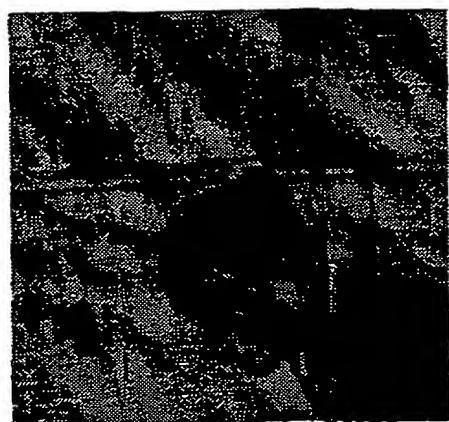
a)
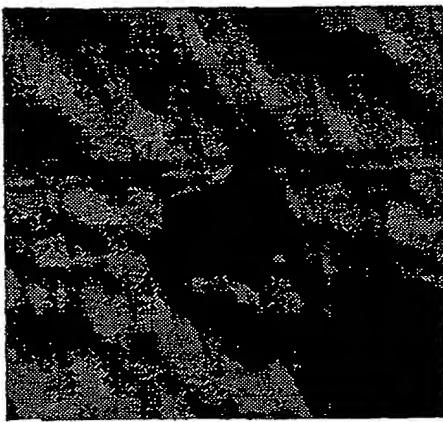
b)
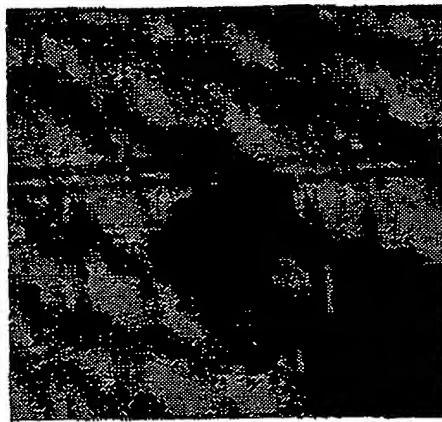
c)
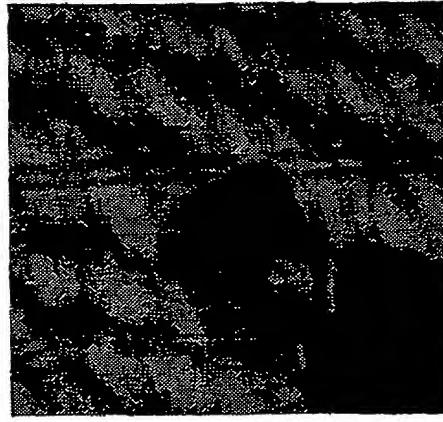
d)

Fig. 11

BEST AVAILABLE COPY